ECET 491

Product Length Measurement and Feedback System for Brake Line Cutter

Presented by Tom Overmyer

Instructor: Prof. Paul I-Hai Lin
Advisor: Dr. David Momoh
December 9, 2016

1

Outline

- Executive Summary
- Introduction
- Problem Statement and Solution
- Key Requirements
- System Design
- System Build and Testing
- Repeatability Issue
- Conclusion
- Video Demonstration

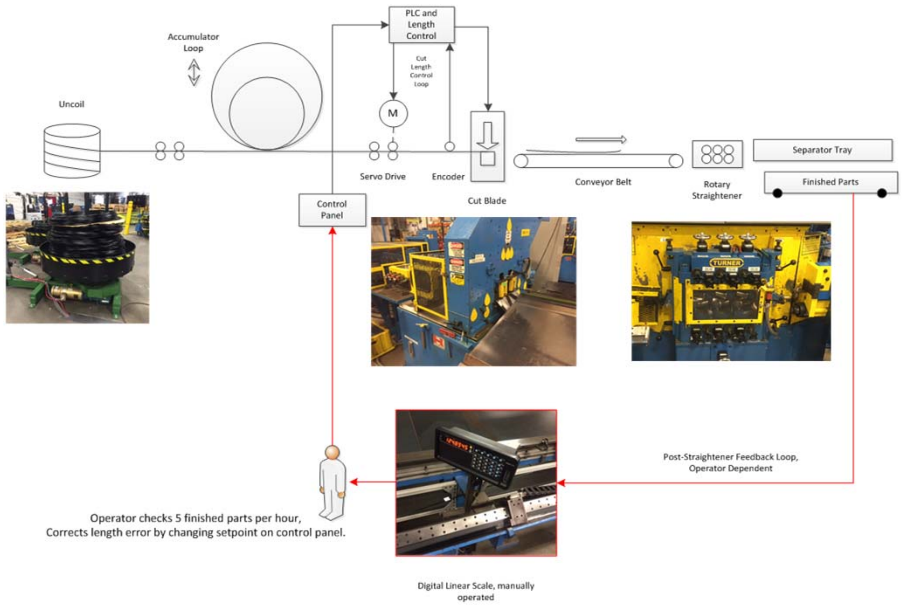
2

Executive Summary

- Motivation
  - Reduce scrap in a brake line cutting process
- Deliverables
  - Prototype
  - Final Report
  - Final Presentation
    - Video demonstration of the prototype
- 15 weeks
- $3000, 573 labor hours

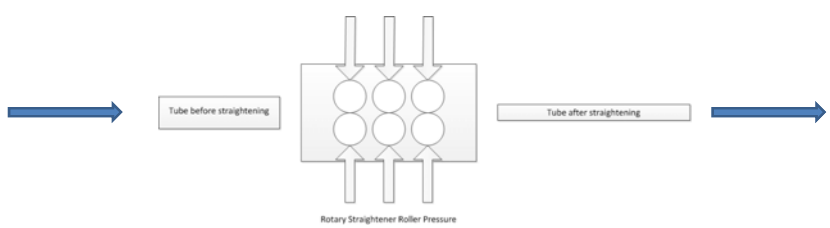
Introduction

- TI Automotive
- Manufacture brake and fuel lines for Ford, GM, and Chrysler
- Steel brake line tubing comes in 30,000 foot coils
- First step in the process is uncoil the tubing, cut to length, and straighten

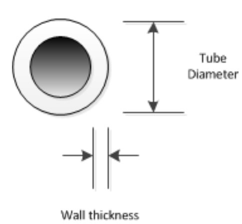


5

The Problem



- Rotary straightener applies pressure to the outside diameter of the tube
- Finished length changes as a result
- Straightener adds 0.1” to 0.2”, but depends on the material



6

The Problem

- The tubing's material properties change often
- The finished length may change between quality checks
- If the material changes drastically, the finished length will be outside of the allowed tolerance
- The operators sort out the scrap parts
- Quality department data shows $8500 in scrap from one machine last year

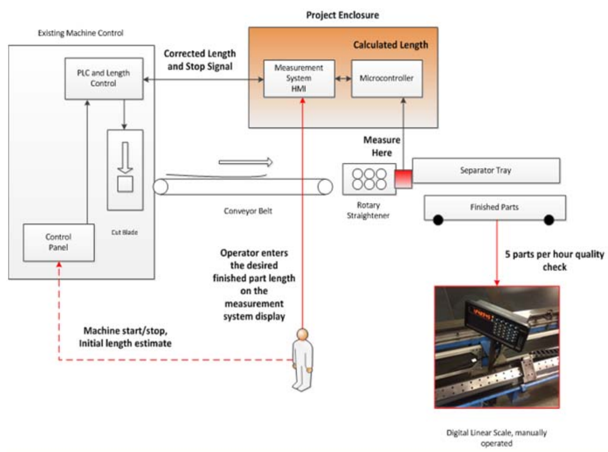
7

The Solution

- Measure the length of every tube after the straightener
- Send length correction back to cutter's PLC
- Stop the cutter if tubes are out of tolerance

8

Project Overview



Requirements

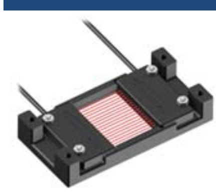
- Measure tubes after they pass through the straightener
- Tubes will range between 15” and 225”
- Tubes 100” or longer within +/-0.062”
- Tubes shorter than 100” within +/-0.031”
- Display measurements, provide user interface
- Communicate with an Omron NJ PLC
- Send corrected length to PLC
- Should be accurate to within 0.005”

Length Measurement

- Tube length = tube speed x time
- Accurate timer
- Speed must be known
- Fiber optic sensor to detect tube, start and stop timer
- Optical encoder to monitor motor speed which is proportional to tube speed

11

Length Measurement



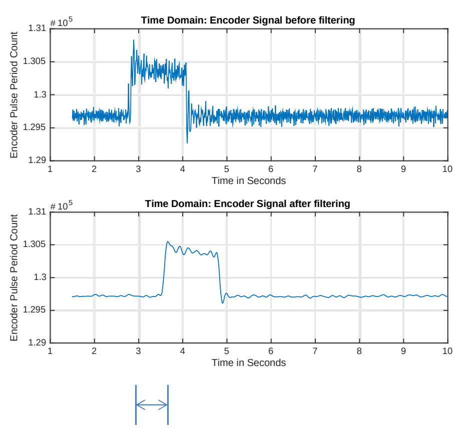
- Fiber optic array sensor to detect tubes
- Banner DF-G2 Amplifier
  - 10µs response time
  - 10 – 30 VDC device
- Arduino Due
  - 42Mhz clock for time measurement (24ns resolution)
  - 32 bit for large timer counts

12

Length Measurement

- Optical encoder for monitoring motor speed
- 1024 PPR
- 10 – 30VDC device

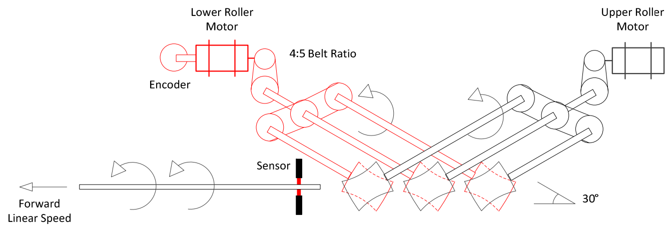
Motor Speed



Filter time delay

- Encoder period measured every 10ms
- Noise is filtered out before converting to RPM
- Matlab used to design a 151 tap lowpass FIR filter
- Delay of 0.75 seconds is accounted for in the Arduino program

Length Measurement



- 30 degree roller angle causes tube to travel in a helical motion
- Forward linear component of motion used in length calculation
- Fiber optic sensor marks beginning and end time of measurement
- Encoder mounted on lower roller motor– measure RPM and then convert to forward linear speed of tube

15

Length Calculation

$$Linear\ tube\ speed\ in\ inches\ per\ second, \quad v = \frac{Roller\ RPM\ \cdot 7.069 \cdot \sin 30°}{60}$$

$$Tube\ Length\ = v\ \cdot time$$

- Roller speed varies, so length is calculated at small intervals and then summed

$$Tube\ Length = \sum_{i=2}^{n}[(v_i \cdot T_s) + (v_1 \cdot T_{start}) + (v_n \cdot T_{end})]$$

$$where\ T_s = Sampling\ period, 10ms$$

$$v_i = linear\ tube\ speed\ v\ during\ each\ sample$$

$$T_{start} = T_s - sensor\ start\ time\ between\ samples$$

$$T_{end} = sensor\ end\ time\ between\ samples$$

16

Length Measurement Summary

- Tube interrupts fiber optic sensor beam
  - Generates interrupt in Arduino that resets a timer
  - Record start time relative to first encoder measurement
- Store motor speed in a buffer, every 10ms
- Tube leaves fiber optic sensor beam
  - Generates interrupt that records stop time relative to last encoder measurement
- Calculate length using recorded motor speed along with start and stop time

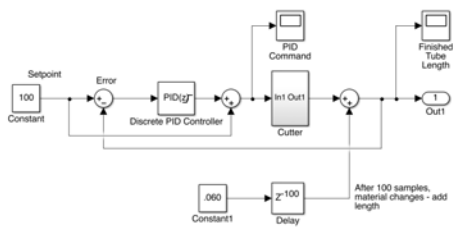
17

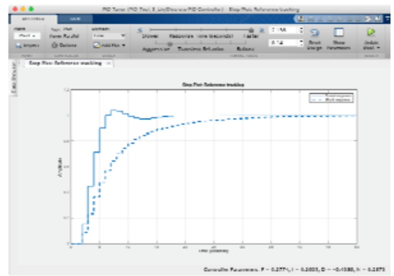
Length Correction Feedback

- Moving average was considered
  - May take a long time to get back to correct length
- PID should be faster
- Non-continuous process
- Analog PID won't work
  - Integral term will run away between measurements
- Adapted a discrete-time PID for use in this project
- Each measurement increments sample time, rather than fixed sample rate
  - PID is "paused" between measurements

18

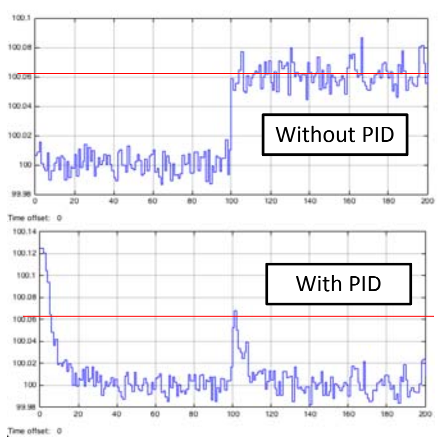
Simulink PID Model





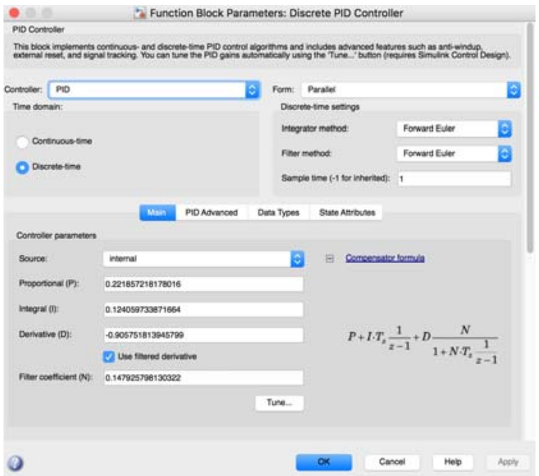
- Simulink provides PID tuning dialog
- This model allowed testing different process shifts
- After 100 samples, add 0.060” to check effectiveness of PID

Simulation Results



- Adding 0.060” results in many out of tolerance parts
- Adding 0.060” results in only one out of tolerance part
- Spike at left is due to the PID step response – accounted for in Arduino program

PID Implementation



Simulink provides a digital transfer function as well as the tuned PID constants

PID Implementation

$$H(z) = P + I \cdot T_s \frac{1}{z-1} + D \frac{N}{1 + N \cdot T_s \frac{1}{z-1}}$$

Transfer function given by Simulink

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}$$

A LOT of algebra

Where:
$b_0 = P + DN$
$b_1 = P(-2 + NT_s) + IT_s - 2DN$
$b_2 = P(1 - NT_s) + IT_s(NT_s - 1) + DN$
$a_0 = 1$
$a_1 = NT_s - 2$
$a_2 = 1 - NT_s$

Inverse z-transform gives the difference equation, implemented in C++ on Arduino

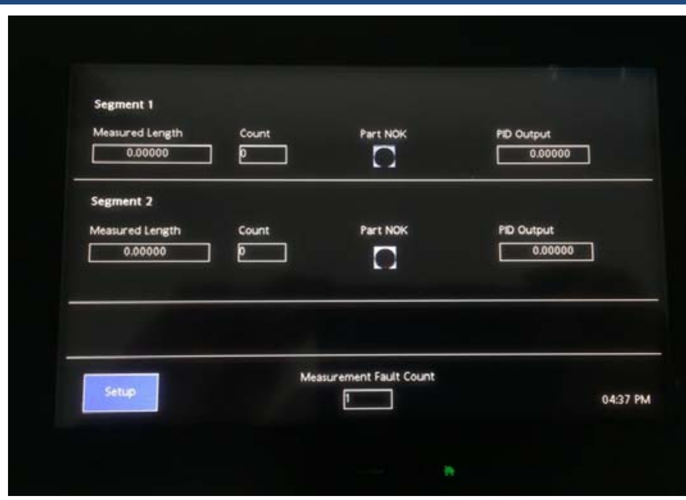
$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2)$$

Communication and User Interface

- Red Lion HMI
- Touch screen user interface
- Communication with over 300 different protocols
- Ethernet to Omron PLC
- Modbus over RS232 to Arduino
- Provides communication link between the Arduino and the PLC
- Ties everything together

23

HMI

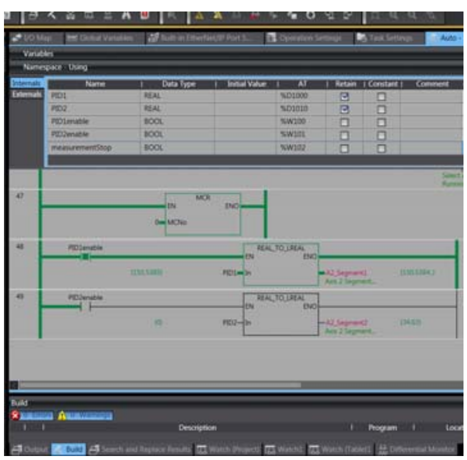


24

HMI

25

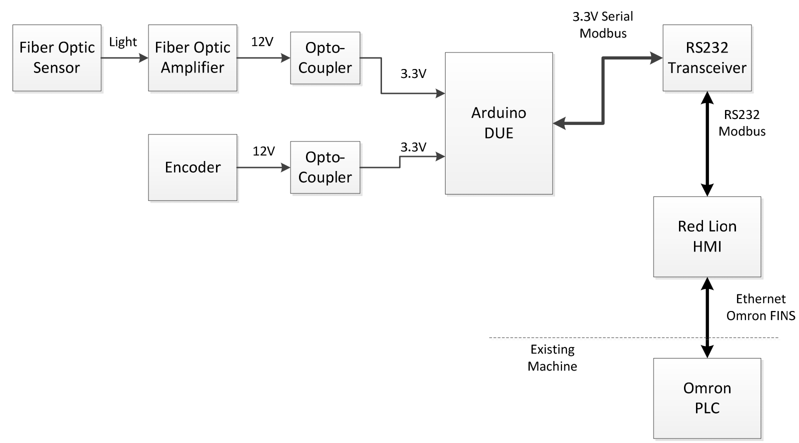
PLC Modification

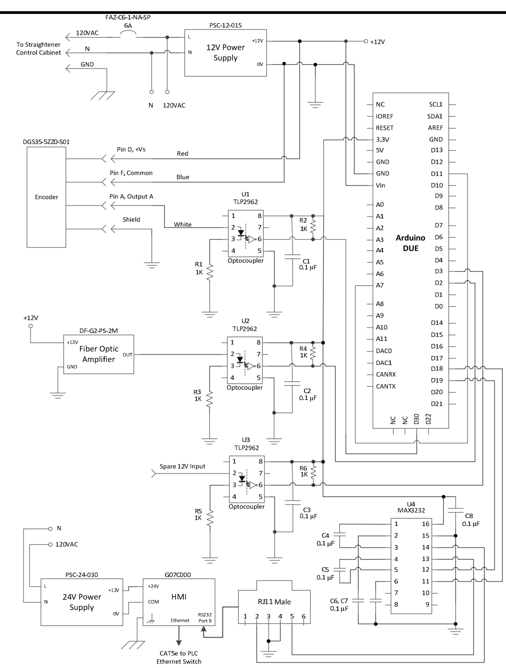


- PID overwrites the tube feed servo's relative motion command
- Replaces manual entry by the operator
- Stop signal sent to PLC if a tube is out of tolerance
- Breaks the Auto Cycle rung that is latched on by the start button

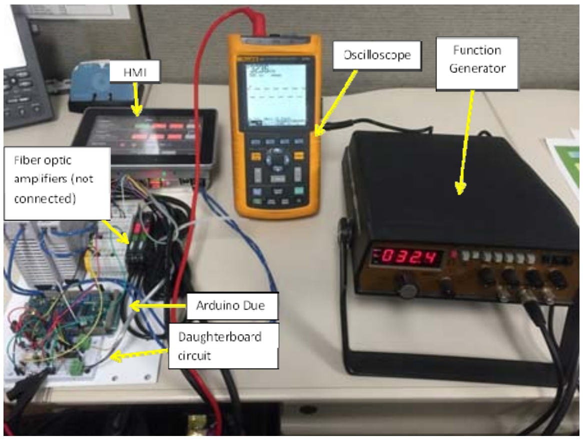
26

Block Diagram





Software Testing



- Verify RPM measurement
- Function generator to simulate encoder signal
- Verify length calculation
- Set encoder variable to fixed value
- Function generator to simulate fiber optic sensor

29

RPM Results

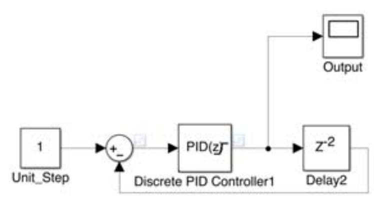
Frequency in Hz	Lowest Expected pulsePeriod	Highest Expected pulsePeriod	Lowest Expected RPM	Highest Expected RPM	Measured pulsePeriod	Measured RPM	Result
100.2	42126802	41707630	5.84	5.90	41925279	5.87	PASS
1001	4216889	4174930	58.36	58.95	4200340	58.59	PASS
5007	843041	834653	291.91	294.85	838675	293.43	PASS
10010	421689	417493	583.59	589.46	419335	586.87	PASS
14990	281595	278793	873.93	882.71	280235	878.1	PASS
20000	211056	208956	1166.02	1177.73	209972	1171.64	PASS
24930	169319	167634	1453.44	1468.05	168512	1460.32	PASS
30010	140657	139258	1749.61	1767.19	139975	1758.19	PASS
35050	120431	119233	2043.44	2063.98	119832	2053.76	PASS
39930	105713	104661	2327.95	2351.35	105154	2339.83	PASS
45030	93740	92808	2625.28	2651.67	93251	2639.05	PASS
49770	84813	83969	2901.63	2930.79	84321	2917.63	PASS
55010	76734	75970	3207.13	3239.36	76327	3223.4	PASS
60090	70247	69548	3503.29	3538.50	69825	3522.12	PASS

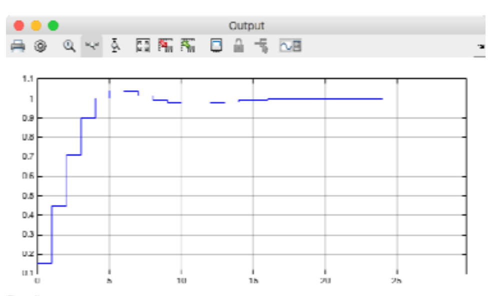
30

Measurement Calculation Results

Measured Time, Seconds	Lowest Expected Length	Expected Length	Highest Expected Length	Calculated Length	Result	% error
0.099	8.770	8.832	8.894	8.786	PASS	0.52%
0.2	17.718	17.842	17.967	17.869	PASS	0.15%
0.299	26.488	26.674	26.861	26.719	PASS	0.17%
0.405	35.878	36.131	36.384	36.083	PASS	0.13%
0.503	44.560	44.874	45.188	44.837	PASS	0.08%
0.604	53.507	53.884	54.261	53.833	PASS	0.09%
0.695	61.568	62.002	62.436	62.053	PASS	0.08%
0.804	71.224	71.727	72.229	71.845	PASS	0.17%
0.908	80.438	81.005	81.572	81.072	PASS	0.08%
1.013	89.739	90.372	91.005	90.332	PASS	0.04%
1.119	99.130	99.828	100.527	99.916	PASS	0.09%
1.206	106.837	107.590	108.343	107.514	PASS	0.07%
1.325	117.379	118.206	119.034	118.197	PASS	0.01%
1.453	128.718	129.625	130.533	129.75	PASS	0.10%
1.576	139.614	140.598	141.583	140.679	PASS	0.06%
1.764	156.269	157.370	158.472	157.376	PASS	0.00%
1.842	163.178	164.329	165.479	164.295	PASS	0.02%
2.01	178.061	179.316	180.572	179.149	PASS	0.09%
2.24	198.436	199.835	201.234	199.825	PASS	0.01%
2.5	221.469	223.030	224.592	223.175	PASS	0.06%

PID Software Testing

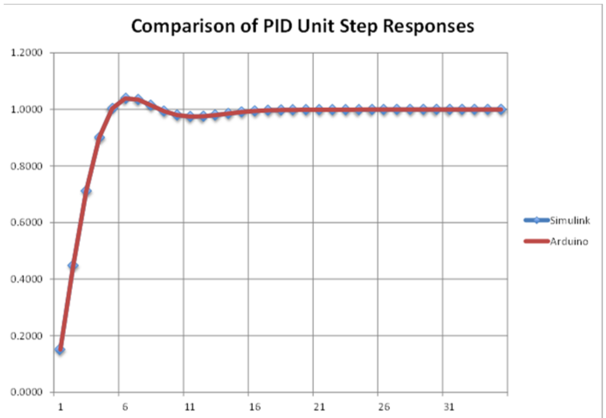




Simulink step response exported and compared with Arduino program step response

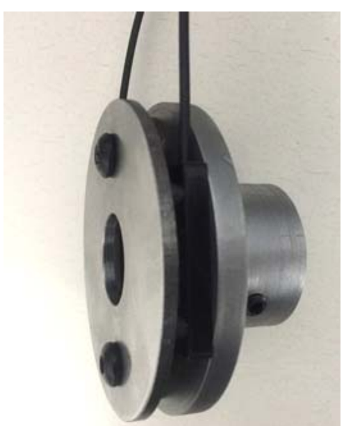
Simulink	Arduino
0.1514	0.151400
0.4481	0.448129
0.7115	0.711541
0.9006	0.900616
1.0030	1.002979
1.0389	1.038878
1.0349	1.034854
1.0147	1.014661
0.9939	0.993876
0.9800	0.979970
0.9744	0.974409
0.9754	0.975378
0.9800	0.979995
0.9857	0.985688
0.9908	0.990779
0.9945	0.994517
0.9968	0.996832
0.9980	0.998027
0.9985	0.998512
0.9986	0.998650
0.9987	0.998686
0.9987	0.998747

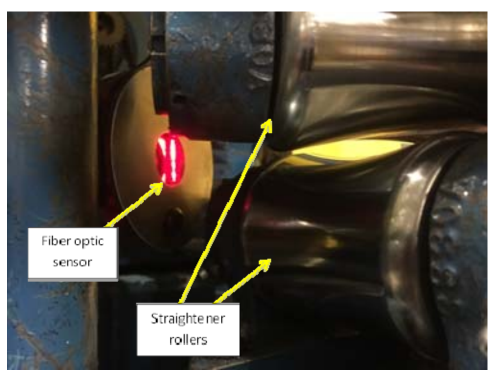
PID Software Testing



Confirmed that the difference equation was implemented correctly

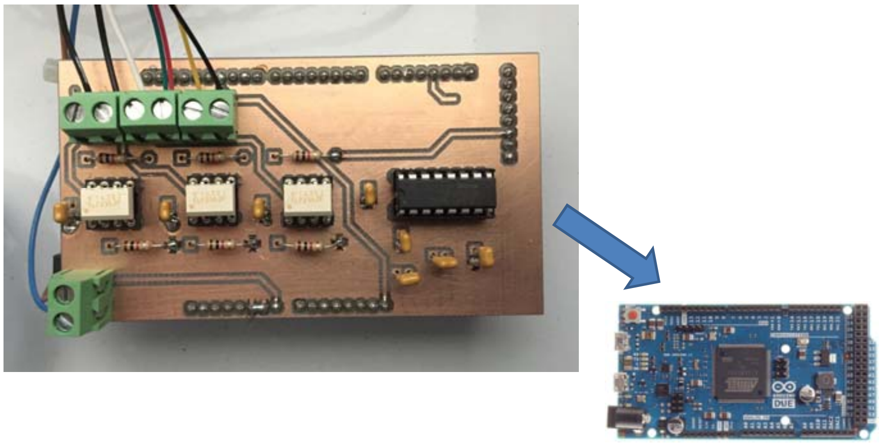
Hardware Build



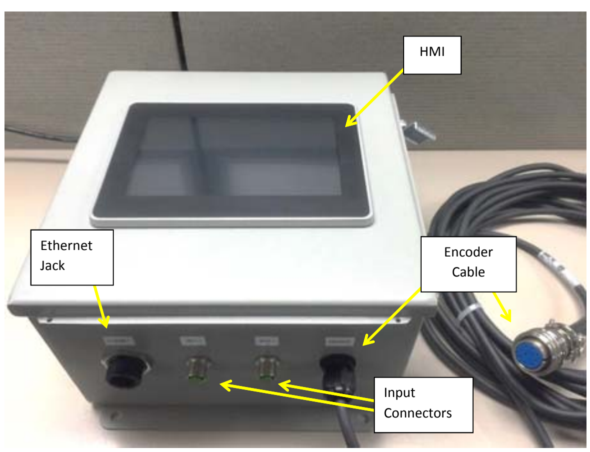


Sensor mount installed

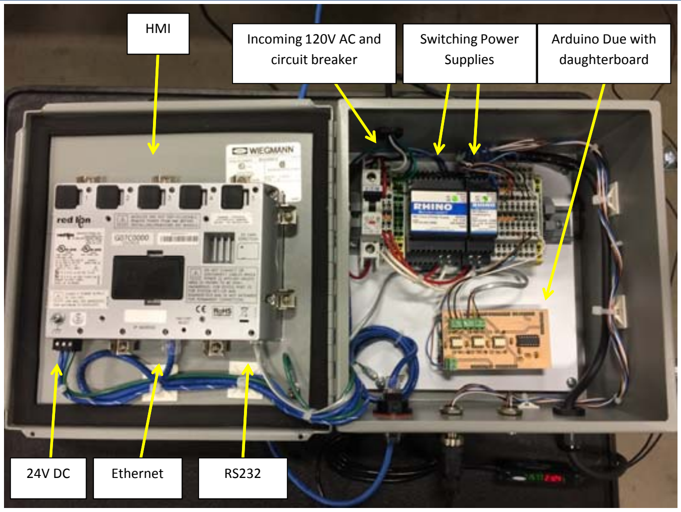
Hardware Build



Hardware Build

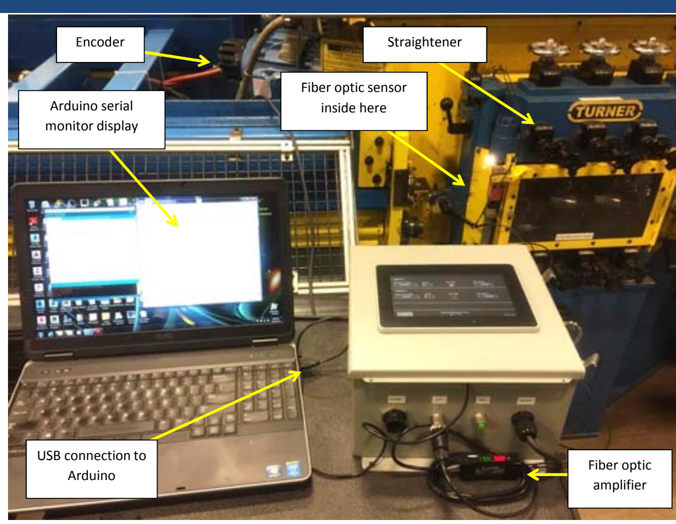


Hardware Build



37

Testing



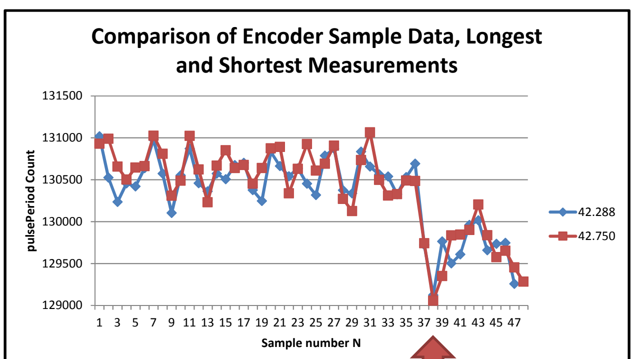
38

Testing Results

Average Actual Tube Length, in inches	Average Measured Tube Length, in inches	Standard Deviation	Range	Number of Tests	FIR Filter	Average Bias, in inches	Estimated Repeatability Error, +- %
42.608	42.713	0.1022	0.462	84	no	0.105	0.47%
42.608	42.709	0.3320	2.496	96	yes	0.101	1.53%
42.608	42.584	0.0978	0.229	7	yes	-0.024	0.45%
34.485	0.000	0.0000	0.000	51	yes	N/A	N/A
34.485	34.238	0.0441	0.188	105	no	-0.247	0.25%
34.485	34.242	0.0471	0.314	401	no	-0.243	0.27%
42.364	42.351	0.0510	0.245	38	no	-0.013	0.24%
42.364	42.363	0.0620	0.340	46	no	-0.001	0.29%
42.364	42.401	0.0610	0.267	43	no	0.037	0.28%
61.212	61.106	0.0449	0.244	138	no	-0.106	0.14%
61.212	62.529	0.2065	0.806	161	yes	1.317	0.66%
72.366	72.408	0.0475	0.299	129	no	0.042	0.13%
72.366	72.403	0.0459	0.239	133	yes	0.037	0.12%

Range is too high. Needs to be less than 0.010”

Tracking Down the Problem



These two tubes were within 0.020”, not 0.462” as measured

Sudden decrease in encoder period – motor speeds up here. 8.9” from fiber optic sensor.

Tracking Down the Problem

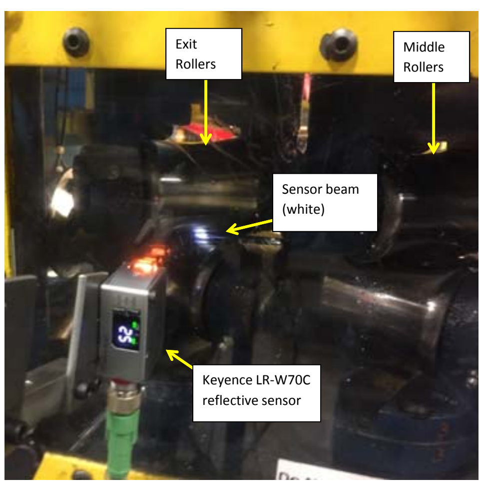
9"

Direction of Travel

Fiber Optic Sensor

Exit Rollers | Middle Rollers | Entrance Rollers

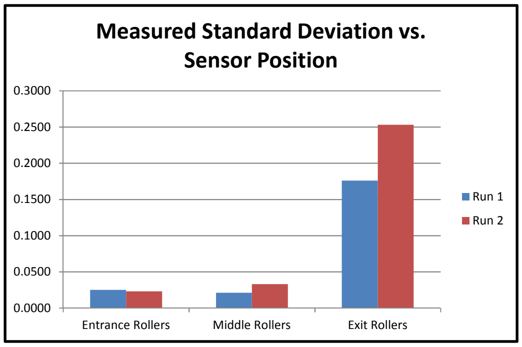
Sensor Position Trial



Temporarily replaced fiber optic sensor with this one

From the Keyence LR-W Series catalog

Sensor Position Trial Results



- Placed reflective sensor to detect tube at different places
- >100 measurements at each position, twice
- Noticeable increase in measurement variation at exit rollers
- Conclusion - tubes are slipping in the exit rollers

Requirements Not Met

- The system should measure tube length with an error of 0.005” or better
  - Within 0.113” at 46”, about 0.24% error
- The system was not permanently installed due to the current level of accuracy

Requirements Met

- Measurements are displayed
- User input functional
- Successful communications with Omron PLC
- Length correction is sent to PLC
- Stop signal is sent to the PLC with out of tolerance measurement

45

Conclusion

- Tubes slipping during measurement
- Tried tightening exit rollers – affects part quality
- Measurement needs to be taken near the middle rollers
- No room to mount the fiber optic sensor there
- Alternative sensor isn't fast enough (250µs)
- More tests with other sensors needed

46

Lessons Learned

- Risk of tube slipping was identified, but likelihood and severity were underestimated
- Thorough risk response should have taken place early in the project
- Communication between Arduino, HMI, and PLC identified as the highest risk
- Proved successful and provided valuable experience for future use

Questions?

Demo